UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/801,792	03/16/2004	Michael A. Holmes	M. HOLMES 3-6	1925

47396 7590 04/29/2008

HITT GAINES, PC
LSI Corporation
PO BOX 832570
RICHARDSON, TX 75083

EXAMINER	
CHOW, CHIH CHING	
ART UNIT	**PAPER NUMBER**
2191	

NOTIFICATION DATE	DELIVERY MODE
04/29/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket@hittgaines.com

Office Action Summary	Application No.	Applicant(s)	
	10/801,792	HOLMES ET AL.	
	Examiner	Art Unit	
	CHIH-CHING CHOW	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *18 February 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *18 June 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on February 18, 2008 has been entered.

2. This action is responsive to amendment dated December 17, 2007.

3. Per Applicants' request, claims 1, 6, 7, 15, and 18 have been amended.

4. Claims 1-20 remain pending.

Response to Arguments

5. Applicant's arguments with respect to claims 1-20 have been considered but are moot in view of the new ground(s) of rejection necessitated by Applicant's amendments to the claims.

6. The examiner has reviewed the updated amendments, and noted that the amendment has changed the scope of the claims, therefore a new prior art has to be introduced. See 35 USC § 103 rejections (claims include the amendments) herein below.

Specification

7. The disclosure is objected to because of the following informalities: Paragraph [0019], recites "The graph generator 110 parses a **High-level Design Language (HDL) file 105**, which may be a **Verilog HDL** file. HDL tools and the files they produce are well known to those skilled in the pertinent art. In the

illustrated embodiment, the **HDL file 105 is produced by a particular hardware description tool** called "OTUS." Of course, other HDL file generating tools fall within the broad scope of the present invention. Table 1, below, illustrates an exemplary OTUS microprocessor interface **hardware description language file that can be the HDL file 105**."; wherein Table 1, recites "Hardware Description Language File" – the "HDL" has two definitions in the Specification, 'High-level Design Language' and 'hardware description language', whereas the Specification should conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention. 'Verilog' is well known as a **hardware description language (VHDL**, see FIG. 1, item 105 and FIG. 2 item 220). "HDL" has two definitions in the Specification, the Examiner is not sure which one does the Applicant really mean, it's assumed to be **hardware description language**. Appropriate correction is required.

Drawings

8. The drawings are objected to because FIG 1. item 105 and FIG. 2 item 220 referred to **VHDL**, which stands for Verilog HDL (see paragraph [0019]). However, in the Specification, the invention really doesn't limit the HDL to be Verilog, instead, it's either 'High-Level Design Language' or 'Hardware Description Language', see Specification objection above. Appropriate correction is required. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

9. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

10. The amended claim 1 and claim 15, recites "a computer-readable medium" which does not appear to be any support in the Specification for this limitation. It does not fix the 35 U.S.C. 101 problem raised in October 17, 2007 Office Action.

11. Claims 1-7 and 15-20 are rejected under 35 U.S.C. 101 as the claimed invention is directed to non-statutory subject matter. In claims 1-7 and 15-20, a "system" is being recited; however, it appears that the system would reasonably be interpreted by one of ordinary skill in the art as software, per se. The only three elements positively recited as part of the system are the "graph generator", "graph converter", and "description generator". It appears that these elements would reasonably be interpreted as representative of the software. As such, it believed that the system of claims 1-7 and 15-20 is reasonably interpreted as software, per se. The deficiency can be fixed by "A system for automatically generating a hierarchical register consolidation structure, comprising: A processor; a graph generator that parses…..".

12. Claims 2-7 depend on claim 1, claims 16-20 depend on claim 15, they do not fix the deficiency of claim 1, and claim 15; therefore they are rejected under 35 USC § 101 for the same reason.

Claim Rejections - 35 USC § 112

13. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

14. Claims 1 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter

which applicant regards as the invention. Claim 1 recites: "a graph generator that parses a High-level Design Language (HDL) file" – wherein the HDL is defined in 'high-level design language' and 'Hardware Description Language' see Specification objection above -- it's not clear to the Examiner what does the HDL really mean in the claim? The Examiner assume it meant to be 'Hardware Description Language' since all the Tables recites 'Hardware Description Language' File, and dependent claim 5 further defined HDL file is produced by a hardware description tool.

15. Claims 2-7 depend on claim 1, they are rejected under 35 USC § 112 (2) for the same reason.

16. Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 7 recites: "The system as recited in Claim 1 wherein said graph generator parses said HDL file in three different passes to generate said intermediate graph,…" – wherein the three different passes each has a different function, see paragraph [0020] of the current application. The functions are important in particularly point out and distinctly claim the subject matter. The claim should add the "first pass is to extract the definitions of all the microprocessor-accessible registers, register names, addresses, bit positions used and their names; the second pass is to identify node interrelationships and associates alarm registers with their mask register and persistency and delta information; and the third pass is to associate summary bits in an alarm register with the alarm register that is summarized and generates bit offsets and masks are generated".

17. Claims 8 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 8 recites: "parsing a High-level Design Language (HDL) file…" – wherein the HDL is defined as 'high-level design language', however the 'HDL' is defined in both 'High-level design language' and 'Hardware Description Language'; see Specification objection above -- it's not clear to the Examiner what does the HDL really mean in the claim? The Examiner assume it meant to be 'Hardware Description Language' since all the Tables recites 'Hardware Description Language' File, and dependent claim 12 further defined HDL file is produced by a hardware description tool.

18. Claims 9-14 depend on claim 8, they are rejected under 35 USC § 112 (2) for the same reason.

Claim Rejections - 35 USC § 103

19. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

20. Claims 1, 2, 5, 7-12, 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,937,190, by Gregory et al., hereinafter "Gregory", in view of US Patent No. 5,146,583, by Matsunaka et al., hereinafter "Matsunaka".

As per claim 1,

- ***(Currently Amended) A computer-readable medium having stored thereon a plurality of instructions, the plurality of instructions including instructions that when executed by a processor cause the processor to implement a computer implemented system for automatically generating a hierarchical register consolidation structure, comprising:***
a graph generator that parses a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers of external devices identified by said HDL file;

Gregory teaches a HDL tool to generate an intermediate graph, see Gregory's Abstract, "A digital circuit is synthesized from a **text description** of a digital system. During synthesis, **a parse tree with parse nodes** is constructed and retained. The relationship between the parse nodes and the circuit elements synthesized from those parse nodes is retained. "And see FIG. 7, and description in column 27, lines 10-11, "The **parse tree** for the VHDL source is constructed in step 4110 of FIG. 7", Also see column 17 under 1.3.1 'Translation Process Overview'.

- ***a graph converter, associated with said graph generator, that selectively adds virtual elements and nodes to said intermediate graph to transform said intermediate graph into a mathematical tree; and***

Gregory teaches a graph converter see column 3, lines 19-21, " A software translator then **converts that description** into generic technology **structures that directly correspond statement by statement** with the designer's description." A converted graph may look like Figure 12, a

mathematical tree, with operators and operands and traversable.

- ***a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description to form a hierarchical register consolidation structure to provide a logical representation of said microprocessor-accessible registers, node interrelationships, summary bits and masks of said external devices in a programming language suitable for use by a device-independent condition management structure.***

Gregory teaches generating a graph parsed from a HDL, and convert the graph into mathematical tree. Greg also teaches 'a logical hierarchical register', see Greg's column 18 line 63 into column 19, line 4, "optimizers generally do not eliminate **registers** and defined memory elements such as latches and flip-flops. The translation process typically creates a part in the initial GTech circuit **for each bit of a register defined by the HDL text**. These initial parts have a one-to-one correspondence with final library parts which are chosen by the optimization process. Therefore, partial analysis results associated with the **register** (such as its area) or nets connected to **the register relate directly to those in the initial GTech circuit**. Furthermore, **the final register can be related back to the HDL which caused it to be created**." -- form a hierarchical register consolidation structure to provide a logical representation of said microprocessor-accessible registers; but Gregory does not teach a description generator associated with the graph converter, however Matsunaka teaches it in an analogous prior art; see Matsunaka's claim 1, "a **parsing step for parsing**, using a computer system, the function description of the circuit in **hardware description language** to **generating a parse tree**; **a tree structure**

deforming step for deforming, using the computer system, a structure of the parse tree generated during said parsing step and for optimizing a text level redundancy in said parse tree; and a **function block generation processing step for translating**, using the computer system, the thus **deformed parse tree into a circuit configuration** made up of **function blocks**, thereby **translating the function description of the circuit into connection information among function blocks** as a hardware image." Further see Matsunaka's column 4, lines 50-54, "ATUOl as used here is not an automaton name, but is instead **a register having reset terminal RSET**. By using such an expression, the transition rule can be expressed by numerically expressing the present state and keeping that state." – description associated with the deformed tree (hierarchical register consolidation structure), as function blocks descriptions. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Gregory's disclosure of parsing HDL into graph by Matsunaka's adding description to the graph. The modification would be obvious because one of ordinary skill in the art would be motivated to optimize the parsed tree and make the translation or logic design to the designer or a knowledge source. (Matsunaka's column 2, lines 18-20).

As per claim 2,

- ***The system as recited in Claim 1 wherein said intermediate graph further contains bit offsets associated with said alarm registers.***

Claim 1 rejection is incorporated, Gregory also teaches "interrupt" (alarm register), see Gregory column 34 last two lines, into column 35, lines 32-40, "The function of this VHDL source code model is to compute whether **a new**

interrupt of a particular priority should be serviced given priority over the interrupt being serviced. A request for a processor interrupt arrives on inputs new_request[3], new_request[2], or new_request[1]. The input current_level[1:0] **indicates the priority level of the interrupt currently being serviced**. If the request for an interrupt comes in on a higher level input than the current level of the interrupt being serviced, then the should_service output is set. Otherwise, it is set low." – the interrupt/alarm register processing is similar to the interrupt/alarm register processing (see paragraph [0007] of the current application).

As per claim 5,

- ***The system as recited in Claim 1 wherein said HDL file is produced by a hardware description tool.***

Claim 1 rejection is incorporated, further see Gregory's column 3, lines 9-19, "Logic synthesis was developed to provide the designer with an automatic mechanism to translate **a hardware description language (HDL) description** of a desired function to a structural description of a digital circuit that performed the desired function. Logic synthesis begins with the designer describing the desired function using VHDL, **Verilog**, or any other logic synthesis source language, to specify the behavior. This allows the designer to specify the digital circuit at a higher level, and allows the **CAD tools to assist the designer in defining the functionality of the digital circuit**."

As per claim 7,

- ***The system as recited in Claim 1 wherein said graph generator parses said HDL file in three different passes to generate said intermediate***

graph.

Claim 1 rejection is incorporated, it's well known to the people in the art that any parsing process can parse HDL multiple times, therefore, parsing a HDL file three different passes is not considered novelty, unless the specific function that is accomplished during each of the passes is included.

As per claim 8,

- ***A method of automatically generating a hierarchical register consolidation structure, comprising:***

parsing a High-level Design Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits and masks associated with alarm registers;

selectively adding virtual elements and nodes to said intermediate graph to transform said intermediate graph into a mathematical tree; and

employing said mathematical tree to generate a static tree description in a programming language suitable for use by a device-independent condition management structure.

Claim 8 is a method claim of claim 1, therefore see claim 1 rejection.

As per claim 9,

- ***The method as recited in Claim 8 wherein said intermediate graph further contains bit offsets associated with said alarm registers.***

Claim 8 rejection is incorporated, claim 9 is the method version of claim 2, therefore see claim 2 rejection.

As per claim 10,

- ***The method as recited in Claim 8 further comprising employing said static tree description to generate an HTML traversable tree representation based on said mathematical tree.***

Claim 8 rejection is incorporated, claim 10 is the method version of claim 3, therefore see claim 3 rejection.

As per claim 11,

- ***The method as recited in Claim 8 wherein said programming language is C.***

Claim 8 rejection is incorporated, claim 11 is the method version of claim 4, therefore see claim 4 rejection.

As per claim 12,

- ***The method as recited in Claim 8 further comprising producing said HDL file with a hardware description tool.***

Claim 8 rejection is incorporated, claim 12 is the method version of claim 5, therefore see claim 5 rejection.

As per claim 14,

- ***The method as recited in Claim 8 wherein said parsing, selectively adding and employing are carried out by sequences of instructions executable in a general purpose computing system.***

Claim 8 rejection is incorporated, Gregory's teaching is for a general

purpose computing system, see Gregory's column 14, lines 18-19, "the design of a digital circuit can be represented or specified in different ways within the memory of a **computer system**"

21. Claims 3, 4, 6, 13, 15-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,937,190, by Gregory et al., hereinafter "Gregory", in view of US Patent No. 5,146,583, by Matsunaka et al., hereinafter "Matsunaka"; further in view of US Patent No. 6,760,888 by Killian et al., hereinafter "Killian".

As per claim 3,

- ***The system as recited in Claim 1 wherein said description generator further generates an HTML traversable tree representation based on said mathematical tree.***

Claim 1 rejection is incorporated, Gregory teaches a HDL tool to generate an intermediate graph, Matsunaka teaches generating descriptions for the graph, but they do not teach 'HTML'. However, Killian teaches it in an analogous art. See Killian's column 10, lines 55-66, "The automated processor generation system 10 may be a stand-alone system running on a computer system under the control of the user; however, it preferably runs primarily on a system under the control of the manufacturer of the automated processor generation system 10. User access may then be provided over a communication network. For example, the **GUI may be provided using a web browser** with data input screens written in **HTML** and Java." – Killian's teaches develop application instructions for a processor generation system based on HDL (see Killian's

Abstract), and it can present the application instruction user interface in HTML. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Gregory's disclosure of parsing HDL into graph and Matsunaka's adding description to the graph by traversing the graph in HTML. The modification would be obvious because one of ordinary skill in the art would be motivated so the user access may then be provided over a communication network. (Killian's column 10, lines 63-64).

As per claim 4,

- ***The system as recited in Claim 1 wherein said programming language is C.***

Claim 1 rejection is incorporated, Gregory teaches a HDL tool to generate an intermediate graph, Matsunaka teaches generating descriptions for the graph, but they do not teach 'HTML'. However, Killian teaches it in an analogous art. See Killian's column 19, lines 63-67, " n addition to selecting preconfigured characteristics of the processors 60, the search algorithms can also be used to automatically select or suggest to the users possible TIE extensions. Given the input goals and given examples of user programs written perhaps **in the C programming language,** these algorithms would suggest potential TIE extensions."

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Gregory's disclosure of parsing HDL into graph and Matsunaka's adding description to the graph by generating C programming language. The modification would be obvious to the people ordinary skill in the art since C programming language is a well known programming language in the art. (See Killian's column 27, lines 34-

46).

As per claim 6,

- ***(Currently Amended) The system as recited in Claim 3 ~~4~~ wherein said condition management structure manages interrupts associated with said external devices employing said static tree or said HTML traversable tree interacts only with a logical representation of said microprocessor accessible registers, node interrelationships, summary bits and masks.***

Claim 3 rejection is incorporated, Gregory teaches a HDL tool to generate an intermediate graph, Matsunaka teaches generating descriptions for the graph, but they do not teach 'HTML' and 'bits and masks'. However, Killian teaches then in an analogous art. For HTML feature see claim 3 rejection. for 'bits and masks' feature See Killian's column 20, lines 47-53, "Alternatively or in conjunction therewith, a compiler-like tool checks if the user program **uses bit-mask operations to insure that certain variables** are never larger than certain limits. In this situation, the tool suggests to the search engine 106 a co-processor 98 using data types conforming to the user limits (for example, 12 bit or 20 bit or any other size integers)."

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Gregory's disclosure of parsing HDL into graph and Matsunaka's adding description to the graph by using bit-mask operations. The modification would be obvious to the people ordinary skill in the art to insure that certain variables are never larger than certain limits. (See Killian's column 20, line 49).

As per claim 13,

- ***The method as recited in Claim 8 wherein said condition management structure interacts only with a logical representation of said microprocessor-accessible registers, node interrelationships, summary bits and masks.***

The claim 8 rejection is incorporated, for bits and masks feature see claim 3 rejection.

As per claim 15,

- ***(Currently Amended) A computer-readable medium having stored thereon a plurality of instructions, the plurality of instructions including instructions that when executed by a processor cause the processor to implement a ~~computer implemented~~ system for automatically generating a hierarchical register consolidation structure, comprising:***

a graph generator that parses a hardware description language N4gh level De-sign Language (HDL) file to generate an intermediate graph containing definitions of microprocessor-accessible registers, node interrelationships and summary bits, bit offsets and masks associated with alarm registers of external devices identified by said HDL file;

a graph converter, associated with said graph generator, that selectively adds virtual elements and nodes to said intermediate graph to transform said intermediate graph into a mathematical tree; and

a description generator, associated with said graph converter, that employs said mathematical tree to generate a static tree description in a programming language suitable for use by a device- independent condition management structure and an HTML traversable tree representation based on said

mathematical tree, wherein both of said static tree description and said HTML traversable tree provide a logical representation of said microprocessor-accessible registers, node interrelationships, summary bits and masks of said external devices.

Claim 15 is a computer-readable medium having stored thereon a plurality of instructions version of claim 1, therefore see claim 1 rejection. For HTML feature see claim 3 rejection.

As per claim 16,

- ***The system as recited in Claim 15 wherein said programming language is C.***

Claim 15 rejection is incorporated, for C language feature see claim 4 rejection.

As per claim 17,

- ***The system as recited in Claim 15 wherein said HDL file is produced by a hardware description tool.***

Claim 15 rejection is incorporated, for the hardware description tool feature see claim 5 rejection.

As per claim 18,

- ***(Currently Amended) The system as recited in claim 15 wherein said condition management structure interacts only with said a logical representation of said static tree description or said HTML traversable tree to manage interrupts from said external devices microprocessor accessible registers, node interrelationships, summary bits and masks.***

Claim 15 rejection is incorporated, for the rest of claim 18 feature see claim 13 rejection.

As per claim 19,

- ***The system as recited in Claim 15 wherein said graph generator, said graph converter and said description generator are embodied in sequences of instructions executable in a general purpose computing system.***

Claim 15 rejection is incorporated, for the rest of claim 19 feature see claim 14 rejection.

22. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 5,937,190, by Gregory et al., hereinafter "Gregory", in view of US Patent No. 5,146,583, by Matsunaka et al., hereinafter "Matsunaka"; in view of US Patent No. 6,760,888 by Killian et al., hereinafter "Killian"; further in view of US Patent No. 6,928,629 by Johnson, hereinafter "Johnson".

As per claim 20,

- ***The system as recited in Claim 15 wherein said hierarchical register consolidation structure pertains to a real-time system.***

The claim 15 rejection is incorporated, Gregory teaches a HDL tool to generate an intermediate graph, Matsunaka teaches generating descriptions for the graph, Killian teaches HTML and bits and masks, but they do not teach 'real time' explicitly. However, Johnson teaches it in an analogous prior art, see Johnson's column 2, lines 8-10, "**Real-time observability** ("RTO") refers to the ability to monitor and capture internal signals **in real time** either on- or off-chip. While

internal signal observability features have been available in some field programmable gate array ("FPGA") architectures and application specific integrated circuits ("ASICs"), they have typically been of limited scope." And column 6, lines 3-6, "An implementation of the invention described herein thus provides system and method for **parsing HDL events to generate a database for enabling real-time observability in an IC**".

It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Gregory's disclosure of parsing HDL into graph and Matsunaka's adding description to the graph, Killian's presenting the description in HTML, by pertains this disclosure to a real-time system. The modification would be obvious to the people ordinary skill in the art for the purpose of enabling real-time observability when parsing HDL events (See Johnson's column 6, line 5).

Conclusion

23. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Miller et al., US Patent No. 6,877,150, discloses a method of designing an integrated circuit using a general purpose programming language can include identifying a number of instances of each class allocated in a programmatic design implemented using the general purpose programming language and modeling the global memory of the programmatic design.

Ho et al., US Patent No. 6,885,983, discloses a computer is programmed with a search preparation tool that automatically scans descriptions of circuitry to generate a graph, and thereafter automatically traverses the graph to generate a

description that is refined, e.g. by eliminating unnecessary circuitry, such as a declared register that is the destination of an assignment statement in a verilog "always" statement. Specifically, search preparation tool automatically creates a parse tree and thereafter traverses the parse tree to create the graph.

Williams et al., US Patent No. 6,631,508, discloses a method and apparatus for developing placement characteristics of a circuit design in conjunction with developing functional aspects of the circuit. In various embodiments, an application programming interface (API) is programmed in a hardware definition language (HDL). The API provides placement directives that can be called from the HDL code that defines functional characteristics of the circuit.

Chen et al., US 2003/0188302, discloses a hardware design, such as a computer or a component of a computer, is often described or modeled in a Hardware Description Language (HDL) such as Verilog or Very High Speed Integrated Circuit Hardware Description Language (VHDL), or a programming language such as C or C++ or other language. For purposes of illustration, Verilog examples will be described herein, though the techniques described are applicable to other HDL languages (e.g., VHDL), structural level netlists, and graphical design representations as well.

Ly et al., US Patent No. 6,609,229, discloses a programmed computer generates descriptions of circuits (called "checkers") that flag functional defects in a description of a circuit undergoing functional verification. The programmed computer automatically converts the circuit's description into a graph, automatically examines the graph for instances of a predetermined arrangement of nodes and connections, and automatically generates instructions that flag a behavior of a device represented by the instance in conformance with a

known defective behavior. The checkers can be used during simulation or emulation of the circuit, or during operation of the circuit in a semiconductor die The circuit's description can be in Verilog or VHDL and the automatically generated checkers can also be described in Verilog or VHDL.

Broughton et al., US Patent No. 7,080,365, discloses a method for compiling a cycle-based design involves generating a parsed cycle-based design from the cycle-based design, elaborating the parsed cycle-based design to an annotated syntax tree, translating the annotated syntax tree to an intermediate form, and converting the intermediate form to an executable form.

Meribout , US 2002/0162097 A1, discloses a designer describes an LSI circuit with a hardware description language (abbreviated "HDL"). The description using this HDL is suitable to a gate in a final design, and an input source code is relatively short even if the final design is logically complicated.

24. The following summarizes the status of the claims:
    35 USC § 101 rejection: Claims 1-7, 15-20
    35 USC § 112 (2) rejection: Claims 1-14
    35 USC § 103 rejection: Claims 1-20

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 8:30am - 5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Wei Zhen can be reached on 571-272-3708. The fax phone number for the organization where this application or proceeding is assigned is

571-273-8300. Any inquiry of a general nature of relating to the status of this application should be directed to the **TC2100 Group receptionist: 571-272-2100**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Chih-Ching Chow/
Examiner, Art Unit 2191
April 22, 2008

/Ted T. Vo/
Primary Examiner, Art Unit 2191